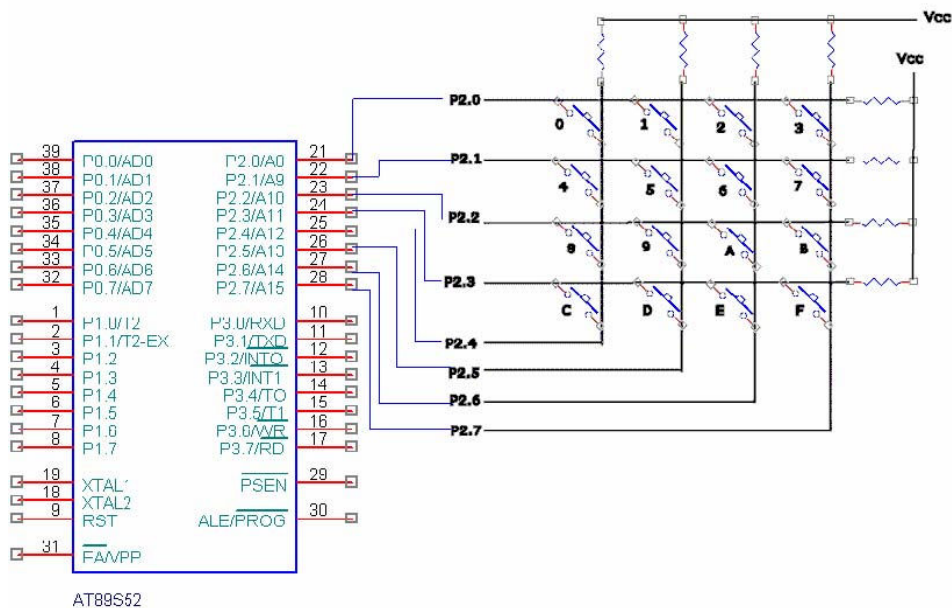


Lista de Exercícios 2

Conceitos envolvidos, além dos desenvolvidos na Lista 1:

- Interface com Teclado Matricial
- Interface com Displays de 7 segmentos
- Interface com LCD
- Interface com Conversores A/D e D/A
- Interface usando Mapeamento de Memória

1) Escrever um programa em Assembly do 8051 que comande dois dispositivos conectados à Porta P1 que controlam uma fechadura para acesso seguro a um ambiente industrial. Somente a senha C42F digitada no teclado matricial, conforme esquema, abre a fechadura através do dispositivo D1 conectado no pino P1.0 quando este bit for setado. Somente a senha F24C aciona o dispositivo D2, conectado ao pino P1.1 que fecha a mesma fechadura quando este bit for setado. Qualquer outra senha digitada zera os pinos P1.0 e P1.1, aciona o alarme A1 conectado ao pino P3.7 setando este bit e aguarda a senha de desbloqueio (369C) que desliga o alarme e permite ao programa voltar ao controle da fechadura.



2) Para o esquema de um Voltímetro com fundo de escala de 15 volts e com resolução de 1 volt construído com o Microcontrolador 8051, considerar os seguintes dados :

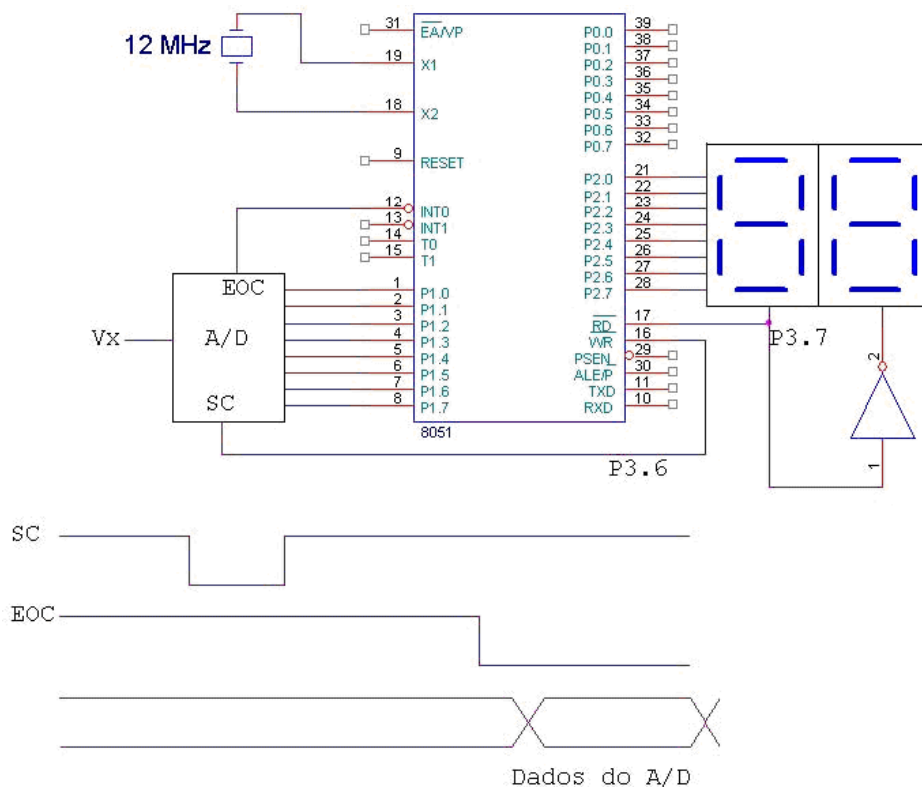
- O Conversor A/D tem tempo de conversão de 100 us (microssegundos) isto é, este é o tempo decorrido entre a subida de borda do pulso SC (Start Conversion) a ser gerado pelo microcontrolador no pino P3.6 e a descida de borda gerada pelo A/D através do sinal EOC (End of Conversion) indicando que os dados estão disponíveis para serem lidos.
- A duração do pulso de SC (Start Conversion) deve ser de no mínimo 50 us (microssegundos).
- Os dois Displays de 7 segmentos são do tipo catodo comum e multiplexados através do pino P3.7 que comanda qual Display será aceso no tempo.

- O valor a ser aceso no Display deve ser disponibilizado na Porta P2 na ordem:



O Pino P2.7 não está sendo usado e cada Display mostrará valores de 0 a 9.

Escrever o Programa Principal que deve converter os dados armazenados no Buffer de Display (endereços da RAM Interna 30h = LSB e 31h = MSB) para 7 segmentos e enviá-los para os Displays Multiplexados, permanecendo continuamente em loop. Programar a Interrupção INT0 para responder ao Conversor A/D. Os valores armazenados no Buffer de Display vão de 00 a 09 para o endereço 30h e de 00 a 01 para o endereço 31h .



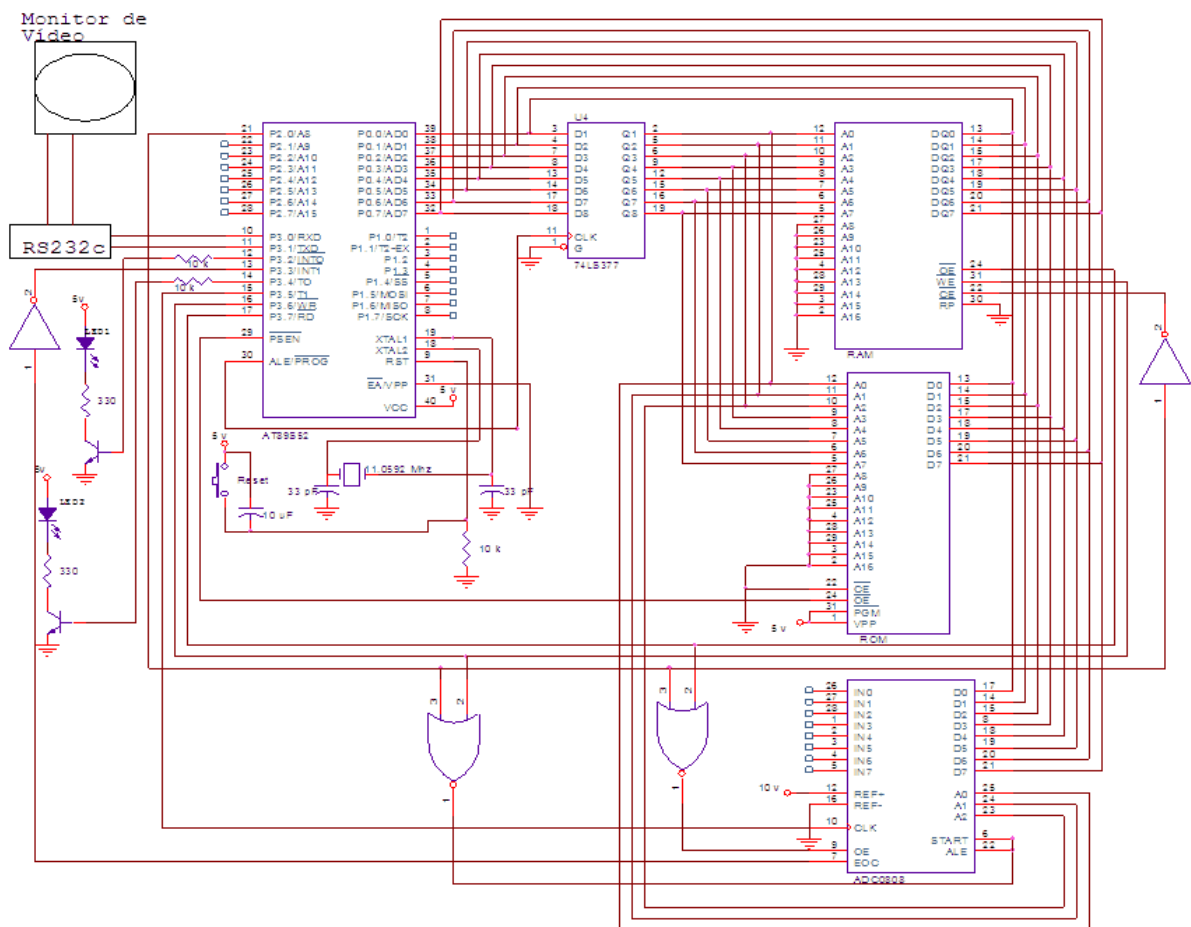
Escrever a sub-rotina de atendimento de Interrupção do Conversor A/D que deverá ler o valor Digital equivalente da Tensão V_x , de acordo com o diagrama de tempos fornecido, e armazená-lo nas posições 30h e 31h da RAM interna (Buffer dos Displays) da seguinte forma:

Valor em P1 – Displays – Buffer dos Displays

De 00 a 0F → 00 volts → 31h = 00 e 30h = 00
 De 10 a 1F → 01 volts → 31h = 00 e 30h = 01
 De 20 a 2F → 02 volts → 31h = 00 e 30h = 02
 De 30 a 3F → 03 volts → 31h = 00 e 30h = 03
 De 40 a 4F → 04 volts → 31h = 00 e 30h = 04
 De 50 a 5F → 05 volts → 31h = 00 e 30h = 05
 De 60 a 6F → 06 volts → 31h = 00 e 30h = 06
 De 70 a 7F → 07 volts → 31h = 00 e 30h = 07
 De 80 a 8F → 08 volts → 31h = 00 e 30h = 08

De 90 a 9F → 09 volts → 31h = 00 e 30h = 09
 De A0 a AF → 10 volts → 31h = 01 e 30h = 00
 De B0 a BF → 11 volts → 31h = 01 e 30h = 01
 De C0 a CF → 12 volts → 31h = 01 e 30h = 02
 De D0 a DF → 12 volts → 31h = 01 e 30h = 03
 De E0 a EF → 14 volts → 31h = 01 e 30h = 04
 De F0 a FF → 15 volts → 31h = 01 e 30h = 05

Na figura abaixo está o circuito de um sistema a microprocessador baseado no Microcontrolador AT89S52 da Atmel mapeado em memória. As questões 3 e 4 referem-se a este circuito.



3) Desenvolver um programa em Assembly que leia um valor analógico no intervalo 0 a 10 v colocado em uma das entradas IN_y do conversor AD ADC0808. (y é um número de 0 a 7)
 Os valores gerados pelo conversor equivalem à seguinte tabela:

0v	---	00
1v	---	17h
2v	---	2Eh
3v	---	45h
4v	---	5Ch
5v	---	73h
6v	---	8Ah

7v --- A1h
8v --- B8h
9v --- CFh
10v --- E6h

O valor da tensão lida no conversor AD deve ser enviado para o Monitor de Vídeo a uma taxa de 9600 BPS, sem paridade e 1 Stop Bit, acompanhado da seguinte informação:

VALOR ANALOGICO DA TENSAO NO SENSOR $y = xx$ volts

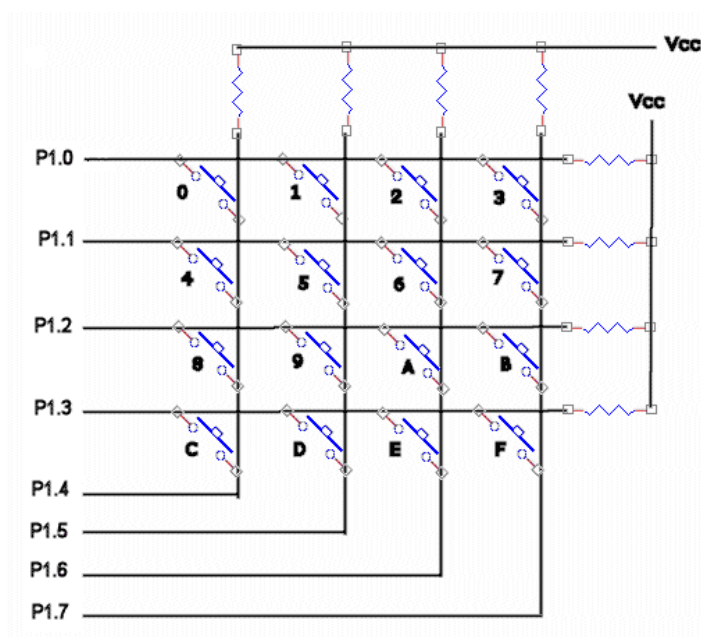
onde xx é um número de 00 a 10 equivalente à tensão na entrada INy e y é um número de 0 a 7 equivalente ao sensor ligado a ela. O programa deve enviar oito valores acompanhados de suas respectivas informações como explicado acima, e ficar em loop para que quando ocorra uma mudança nos sensores, esta mudança apareça diretamente no Monitor de Vídeo.

O Clock do conversor deve ser gerado pelo Timer 0 e deve ser de aproximadamente 50 KHz.

Exemplo da tela do Monitor de Vídeo:

VALOR ANALOGICO DA TENSAO NO SENSOR 0 = 05 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 1 = 09 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 2 = 00 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 3 = 03 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 4 = 10 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 5 = 01 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 6 = 04 volts
VALOR ANALOGICO DA TENSAO NO SENSOR 7 = 08 volts

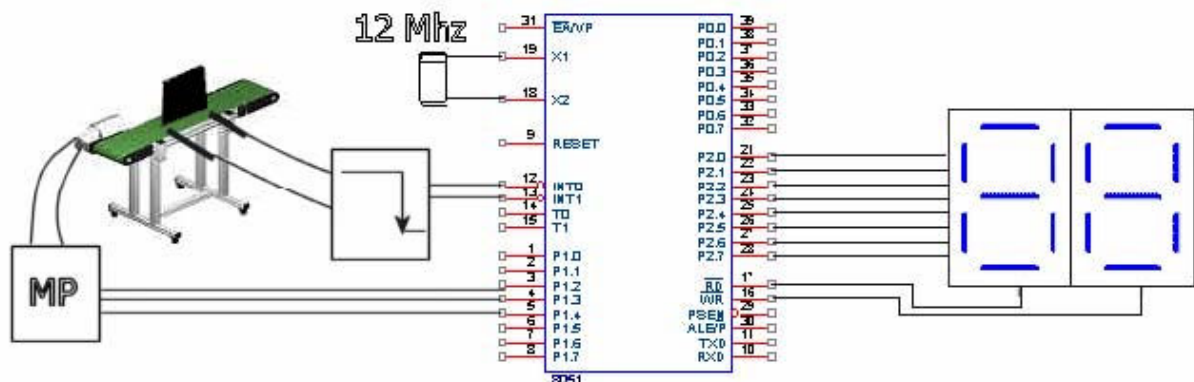
4) Conectar um teclado matricial conforme a figura, na Porta P1 do Microcontrolador.



Escrever um programa em Assembly que permita a um usuário digitar uma senha de 4 dígitos no teclado. Se a senha digitada for 4AD3 em hexadecimal, o programa deve acender o LED1 (P3.2) e preencher a RAM externa tal que o conteúdo de cada posição seja igual à parte menos significativa do endereço. Se a senha for CABE em hexadecimal a RAM externa deve ser totalmente zerada e o LED2 (P3.4) aceso. Qualquer outra senha que for digitada diferente das duas anteriores, o programa deve preencher a RAM externa com FFh, piscar os dois leds de maneira alternada (enquanto um está aceso o outro está apagado e vice-versa) em uma frequência aproximada de 10 Hz e não permitir a entrada de nenhuma outra senha.

Obs: Os Leds inicialmente devem ficar apagados.

5) No esquema um Microcontrolador baseado no 8051 verifica o tamanho de cada peça sobre uma esteira transportadora. A esteira é movimentada através de um Motor de Passo (MP) sendo que cada passo no motor corresponde a 1 mm (milímetro) de deslocamento linear da esteira. Os pinos de comando do MP são: P1.2 = clock, P1.3 (dir) = 1 move a esteira da esquerda para a direita e P1.4 = pára o MP se igual a zero. Na figura, a peça sobre a esteira desloca-se da esquerda para a direita e é colocada na mesma antes do primeiro sensor.



O primeiro sensor, conectado em INT1, deve ser usado para medir a largura da peça em milímetros (de 1 a 9 mm) que deverá ser mostrada no Display de 7 segmentos (catodo comum) mais à direita, somente após a mesma passar completamente pelo segundo sensor ligado em INT0. Este segundo sensor deve contar o número de peças medidas (de 1 a 9) e o valor mostrado no primeiro Display (catodo comum) à esquerda. Os dois sensores devem ativar as interrupções equivalentes na descida de borda quando o objeto sai da frente do sensor. Quando o objeto está na frente do sensor o valor nos bits de porta equivalentes a INT0 = P3.2 e INT1 = P3.3 é igual a um.

a) Escrever o Programa Principal em Assembly que continuamente:

- Comande o Motor de passo a uma velocidade de aproximadamente 200 Hz. O motor não deve ser parado movendo a esteira continuamente.
- Avalie os sensores e meça a largura da peça através do sensor conectado em INT1.

ATENÇÃO: Os Buffers somente serão atualizados pela Interrupção INT0.

b) Escreva as duas Sub-rotinas de Atendimento de Interrupção em Assembly da seguinte forma:

Atualize os dois Displays. O LSB (Display à direita) com o tamanho da peça (de 1 a 9 mm) e o MSB (Display à esquerda) com o número de cada peça (1 a 9). Os dois valores devem aparecer simultaneamente (multiplexados pelos pinos P3.6 para o LSB e P3.7 para o MSB). Quando nenhuma peça tiver sido medida deve aparecer 00 nos dois Displays. Utilizar como Buffer de Display os endereços 30h = LSB e 31h = MSB da RAM Interna. O valor a ser aceso nos Displays deve ser disponibilizado na Porta P2 na ordem:



Onde P2.7 = 0 e P2.0 = g

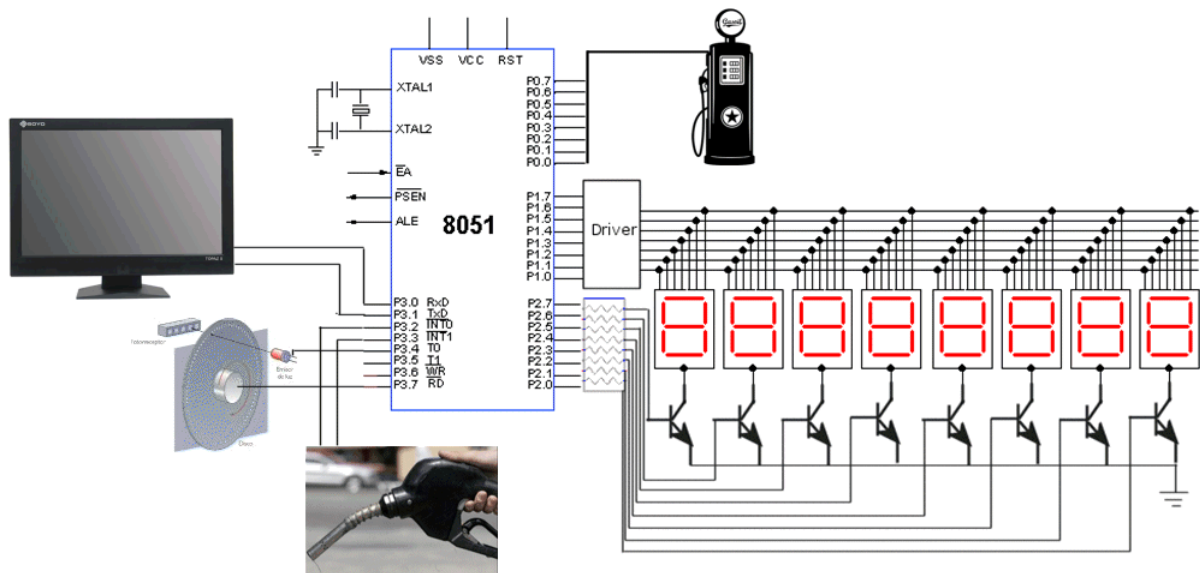
A sub-rotina da INT0 deve atualizar os dois valores (de 1 a 9) nos Buffers (30h = LSB largura da peça e 31h = MSB número da peça).

As duas sub-rotinas devem retornar ao Programa Principal para contínua atualização dos Displays.

A sub-rotina da INT1 deve apenas parar a medida da peça, consolidar o valor em um Registrador intermediário e não deve atualizar o Buffer 30h o que deverá ser feito apenas no final do processo (quando a peça terminar de passar pelo segundo sensor solicitando a interrupção INT0).

- Fornecer o programa em Assembly
- Os parâmetros de programação dos tempos envolvidos

A figura abaixo é o circuito microprocessado de uma bomba de combustível e deverá ser usada para as questões 6, 7 e 8.

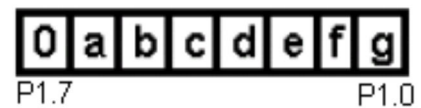


Operação da bomba:

- O motor que bombeia o combustível é acionado pelo pino P3.7 = 1
- Cada litro que for bombeado equivale a um pulso na entrada T0
- Para ligar a bomba, o frentista aciona o gatilho que está ligado na Interrupção INT1, sensível a nível baixo (Baixa prioridade).

- Um sensor instalado no bico do alimentador de combustível está conectado na interrupção INTO, sensível à descida de borda, que detecta quando o tanque está cheio (Alta Prioridade).
- O display de 7 segmentos, catodo comum e multiplexado, mostra nos quatro dígitos menos significativos, acionados por P2.0 a P2.3, a quantidade de combustível transferida, e nos quatro dígitos mais significativos (P2.4 a P2.7) o valor em Reais (considerar 1 litro de combustível = 1 Real).
- Um terminal RS232 (9600, N, 8, 1) instalado no interior do posto de combustível é utilizado para a cobrança.
- A Porta P0 do microprocessador serve para a comunicação de dados com a bomba de combustível.

6) Considerando-se o display de 7 segmentos acionado pela Porta P1, escrever um Programa em Assembly que mostre nos Displays, a quantidade de combustível e o preço (em BCD) durante o tempo de enchimento do tanque. Observar que o motor da bomba é ligado (P3.7=1) quando o gatilho for acionado e desligado (P3.7=0) ou pelo sensor no bico do alimentador (INT0) ou pelo gatilho (INT1=1). Ao ser ligado o motor da bomba, os displays devem ser zerados e ao ser desligado os valores devem permanecer acesos no Display.



7) O preço a pagar e a quantidade de combustível bombeado devem ser transmitidos em ASCII para o terminal RS232 instalado no interior do posto no final de cada operação, ou seja, quando o motor da bomba for desligado. O Microprocessador desabilita as interrupções e aguarda um código de “transação OK” enviado pela linha serial (3Ch) do terminal, retransmitindo-o para a bomba de combustível através da Porta P0 e liberando a bomba para uma nova operação. A liberação da bomba deve ser realizada através da reabilitação das interrupções.

O cristal do microprocessador é de 11.0592 MHz.

Escrever este programa na forma de uma Sub-rotina e indicar em qual ponto no programa do item 1 ela deverá ser chamada.

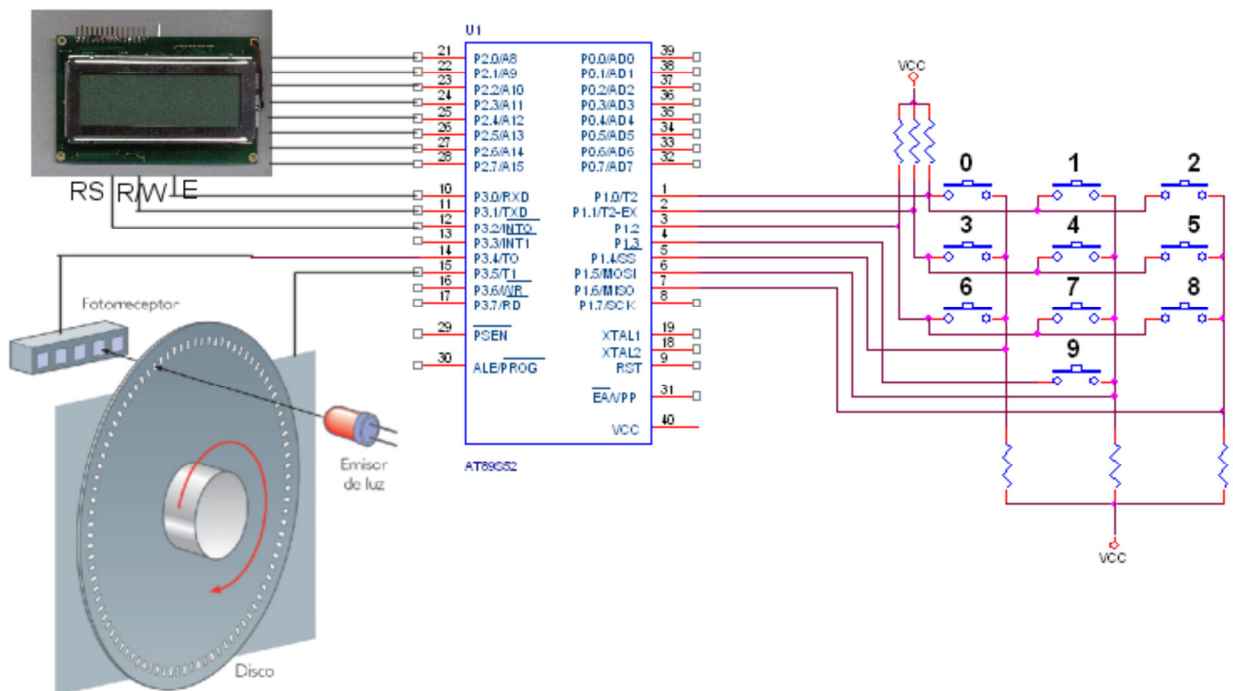
8) Se a bomba de combustível tiver uma memória RAM comunicando-se com a porta P0, seria possível enviar o código 3Ch correspondente à “transação OK” através da instrução

MOVX @R0,A fazendo-se A = 3Ch e R0 = 00

somente com a configuração de hardware apresentada? Se a resposta for negativa, quais pinos extras do 8051 deveriam ser utilizados para tornar este comando viável? E como deveria ser o circuito (hardware) para isso?

Obs: Utilizar apenas pinos do 8051 disponíveis no circuito da Figura, mas, pode-se utilizar hardware adicional.

9) A figura é parte do controle de uma máquina que possui um Motor e um Encoder Óptico diretamente acoplado ao seu eixo. Cada vez que a luz do emissor de luz passa por um dos furos do disco do Encoder, um pulso é gerado no Fotorreceptor ligado no pino T0. Desenvolver um programa em Assembly que entre com a quantidade de pulsos através do Teclado conectado na Porta P1, ligue o Motor (P3.5 = 1) e quando a quantidade de pulsos gerada pelo Encoder for igual ao valor inserido pelo Teclado, enviar uma mensagem ao Display de LCD (16 x 2) com a informação "FERRAMENTA POSICIONADA", parar o Motor (P3.5=0) e aguardar novos dados de posicionamento pelo Teclado.



Atenção:

- 1) Para o Teclado: Utilizar apenas dois dígitos, ou seja, de 00 a 99 para os dados. Matriz → P1.0 a P1.3 são as linhas (Entradas) e P1.4 a P1.6 são as Colunas (Saídas). Não há a necessidade de eliminar ruído de chave.
- 2) Para o LCD: E = P3.0, R/W = P3.1, RS = P3.2